Investigating Call Drops with Field Measurements on Commercial Mobile Phones

Messina, Alessandro; Caragea , Gabriel ; Compta , Pol Torres ; Fitzek, Frank; Rein, Stephan Alexander

Published in:
Vehicular Technology Conference (VTC Spring), 2013 IEEE 77th

DOI (link to publication from Publisher):
10.1109/VTCSpring.2013.6692604

Publication date:
2013

Document Version
Early version, also known as pre-print

Investigating Call Drops with Field Measurements on Commercial Mobile Phones

Alessandro Messina, Gabriel Caragea, Pol Torres Compta, Frank H.P. Fitzek, Stephan A. Rein
Aalborg University, Department of Electronic Systems
Email: {ing.amessina@gmail.com, gabriel.caragea@yahoo.com, pol.torres@gmail.com}, {ff, sr}@es.aau.dk

***Abstract*—One of the biggest problems nowadays for network operators are occurring call drops. This problem has been increasing in the last years specially since the advent of 3G. The investigation in the operator's network is very time intensive and due to the highly priced hardware only a few measurements can be done per day. In this paper we present a new methodology to investigate call drops by using mobile phones to do the measurements following the concept of *citizen sensing*.**

Therefore, a mobile application for Android is made that collects all necessary data and dumps the measurement results in a centralized database where the measurements are evaluated and represented on Google Maps. With a post analysis of the measurements, a classification of the call drops results is made. The collected data is also used to show some statistics related to the battery level and the received signal strength between mobile phone and network. This low cost variant of field testing is developed for call drops but could be used for any other parameter of interest.

***Index Terms*—field data capturing, smartphone, call drops, Android, KPI.**

I. Introduction

Customer experience, user perceived Quality of Service (QoS) or user Quality of Experience (QoE), are key factors of business success. In the telecommunications industry this is not an exception. [1] Some of the reasons why a user experience problems, while using the device, can be related with issues like registration and call setup, call drop, poor throughput or corrupted data.

Since call drops are one of the basic KPIs (Key Performance Indicators) [2] [3] that manufacturers are looking at when they are testing a device, the main focus in this paper is the call drops detection. The aim will be to try understand where does the call drop issue come from? The network or the device?

Next generation smartphones are very complex data centric devices, where the real life performance can be tested only via field tests. This is happening because the performance is dependent upon multiple factors such as: network operator setup, radio environment, number and types of handset antennas, the user type and mode, the handset RF (Radio Frequency), baseband, algorithms and software protocols.

Drive test is very time consuming and not only, it involves having a special vehicle with specific, and usually expensive, equipment and dedicated engineers. The main idea is to find a cheaper way to automate the field data capturing process. Therefore, an Android app (application) was developed that collects field data conveying it to a remote database for later analysis.

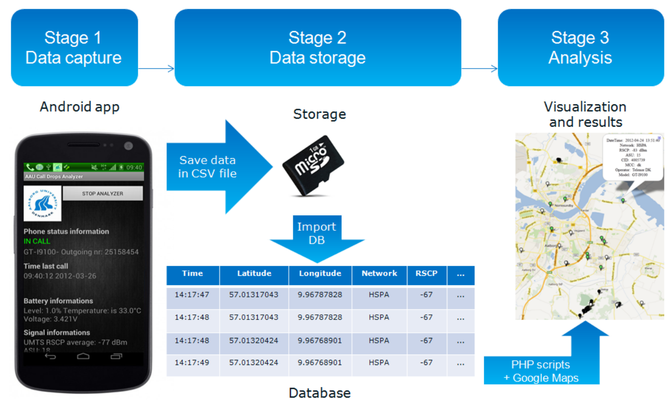

Fig. 1. System overview.

The types of data that can be logged on the phone and transferred to the database are: GPS (Global Positioning System) coordinates, received signal strength, throughput, CID (Cell IDentification), different network parameters, battery consumption, speed and many more. The performance indicators are logged together with the GPS position and afterwards, they can be visualized on Google Maps. An overview of this idea is available in Figure 1.

There have been done several researches related to field data capturing, e.g. [4] which is a mobile app on Google Play Store able to build a worldwide street map and show the status of the current traffic. This is done by collecting individual data from mobile phones carried in cars. In [5], the purpose is to create a database of cell phone towers, cell phone signal strength readings and Wi-Fi access points around the world. Furthermore, [6], [7], [8], [9], [10], are research projects related to field data collection using different scenarios, but none of them is tracking call drops. Other studies have been developed based on the call drops probability using different scenarios, [11], [12], [13], or call drop classification in live GSM (Global System of Mobile communications) networks [14], [15].

The development of this system will bring benefits for companies and network operators. A high level approach using the concept of swarm intelligence is considered. This concept reflects a large number of simple, robust, autonomous devices able to perform together complex tasks, reducing cost and getting more realistic measurement results. Building an app

978-1-4673-6337-2/13/$31.00 ©2013 IEEE

that can detect call drops and problematic areas with respect to this, will give the manufacturers an early and low cost warning of what is coming.

In this paper we present the results obtained with a mobile application able to detect call drops. During the measurement campaign, several mobile phones from different providers are used. These models are Samsung Galaxy SII, Samsung Galaxy Nexus, Samsung Nexus S, Sony Ericsson Xperia Arc, HTC Sensation Z720 and LG Optimus 2X P990.

In order to keep the privacy about their performance we are going to refer inside the paper as anonymous devices. (E.g.: device1, device2, device3, etc.). With a post analysis, a call drop classification is made.

The rest of the paper is organized as follows. Section II describes the measurement testbed in three directions. Section III presents the measurement results, while Section IV outlines a discussion on these results. Finally, Section V concludes the paper and Section VI propose new features that can be added in the future.

II. MEASUREMENT TESTBED

The app developed is for smartphones using Android OS because. This app is able to capture field data, store and upload them on a server to be shown on Google Maps.

We consider three aspects for the testbed of this system: client, server and measurement campaigns.

A. Client

The app *AAU Call Drops Analyzer* collects field data like:

- GPS parameters - latitude, longitude, speed and accuracy;
- Signal parameters - network type, signal strength, asu (android signal unit), LAC (Location Area Code), CID and BER (Bit Error Rate);
- Internet performance - download speed, upload speed and latency (throughput);
- Device parameters - IMEI (International Mobile Equipment Identity), model, time, battery level, battery temperature and battery voltage;
- Phone call numbers - Incoming number and outgoing number;
- Other parameters - MCC (Mobile Country Code), operator, accelerometer sensor, compass sensor, magnetometer sensor and proximity sensor.

Once the app is started, all the data is collected only during the phone call with a sample of $500ms$ and stored in the SD card as a CSV (Comma Separated Value) file. This sample value can be modified if necessary in order to collect more or less data, but lower values were overloading the CPU of the phones.

Involving people to use this system is relevant for obtaining a lot of measurements fast and easy. For doing this, it is necessary to implement in to the application a concept called *citizen sensing* [16][17]. This is a concept where volunteer users are running the app on their devices and share the information in order to reach the goals of the project.

B. Server

The remote database has been configured in a server of Aalborg University using MySQL database. The CSV file can be imported directly from the device, using a dedicated script or manually to the related table in the database. Since the app is saving data only during a call to a fixed line phone, we are saving an internal value to a variable called *inCall*. This variable has a value of -1 every time the call ends.

The parameters that are tracked to establish the reason behind a call drop are: battery level, RSCP (Received Signal Code Power), network, CID.

About the types of call drops that can be detected, the following methods are used:

- Call drops due to bad received signal strength, if the RSCP value is very small and the call was ended, a threshold is present at -100 dBm. The RSCP is the received power measured on a particular physical communication channel. If the call drops with a RSCP value smaller than this threshold, the call drop is categorized as “bad signal strength”.
- CID value comparison, between the moment when the call dropped and the CID of a measurement taken 5 seconds before. If the CIDs are different, then the call drop is categorized as “bad handover” after CID change, regardless of the received signal strength value.
- Hard handover failure, the network failed to change from $3G$ to $2G$, or the network changed to $2G$ but the UE (User Equipment) is stuck and cannot switch to $3G$ again. The same principle explained before is applied, but comparing the network types, not the CIDs, and the call drop is classified as “bad handover”.
- Other types of call drops obtained, they are categorized as “unknown reason”.

Another case is taken into account to not confuse it with an “unknown reason” call drop. If the battery level is less or equal than 1% and the call was ended, then clearly the battery dropped. This is not considered a call drop, since the user can avoid this problem.

An XML file (Extensible Markup Language) is generated from the database, and some PHP scripts (Hypertext Preprocessor) are used to print the data achieved with the app, along with the locations, on Google Maps.

The main point on using Google Maps is to locate the critical regions where call drops occur and also to track how the different parameters vary while the position of the UE is changing.

C. Measurement Campaigns

In order to achieve great amounts of data and prove our system, it was necessary to perform a drive test around Aalborg.

During the drive test six different smartphones were used to capture the data with the app. Four of this phones (device1, device2, device3 and device4) were configured to work just with $3G$ network while the other two phones (device5 and device6) were configured to switch automatically between $3G$ and $2G$.

The first drive test campaign lasted two days and was performed on workdays. The first day test was performed with a sunny/cloudy weather while the second day was raining during all the journey. The second drive test campaign lasted also two days and was performed on workdays. In both days the drive test was performed with a sunny/cloudy weather. A total of 541 kilometers were driven during the four days and a full coverage of Aalborg main streets was performed.

In order to be able to detect call drops, and be sure that those were not normal ended calls we prepared a practical scenario. We did a conference call with all mobile phones calling to a fixed line of our office, so we could be sure that if a call ended was because a failure occurred in the region where we were using our devices and not because of the other participant.

III. MEASUREMENT RESULTS

In this section, the measurements and the results obtained through the drive test are presented. In Figure 2 it is shown the complete route measured with one of the phones.

A total of 44 call drops were found during the drive test and the position of each one is shown in Figure 3. Table I describes the reason of some of the call drops and the parameters that support our argument (marked boxes). Table I shows a short version of the full table of call drops detected in order to make it clear for the reader. In this Table, *Network1* and *CID1* are the Network and the CID respectively from the moment when the call ended, while *Network2* and *CID2* are the same parameters with values taken 5 seconds before the call ended. In Figure 6 we show the histogram of the detected call drops per each device and the classification of the different call drop types.

While looking for call drops, the app is tracking all the other parameters, so other results can be obtained from this huge amount of data acquired, like the variation of the battery, the signal strength, the temperature, etc. Some Matlab scripts were realized for the benchmark of the battery for the devices in $3G$. We are just comparing the 4 phones working in $3G$ as it is known that the phones using the GSM signal can save more battery. The results are showed in Figure 5.

Afterwards, we decided to compare the variation of the signal strength for the different devices used. As explained before we divided the phones in two groups, the ones stuck in $3G$ networks (Figure 4) and the ones that could change automatically from $3G$ to $2G$. We noticed that these two last phones, once they changed from $3G$ to $2G$ while a voice call was in process, they never switched back to $3G$, so almost all the data is from GSM networks.

IV. DISCUSSION

Taking into account the special scenario that we designed, we can state that our system can detect, locate and classify all the call drops occurred.

However, as Figure 6 shows, many times the reason for the call to drop is not clear with the high level parameters that the Android APIs (Application Programming Interface) provide

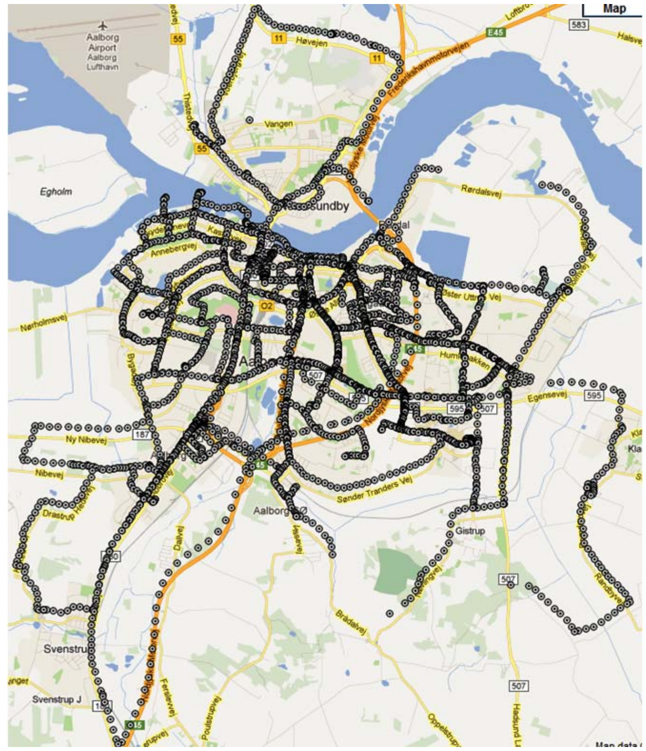

Fig. 2. Map with all the data captured during the drive test with one of the phones.

Fig. 3. Google Map with the position of the call drops.

Call drop reason	Time	Speed (km/h)	Signal strength	Network1	Network2	CID1	CID2	Battery level
Bad signal	13:54:47	0	-113 dBm	UMTS	UMTS	4032050	4032050	75 %
Other reason	13:54:55	55,8	-95 dBm	UMTS	UMTS	4057270	4057270	56 %
Bad handover	13:55:15	40,5	-107 dBm	UMTS	UMTS	4047971	4057270	85 %
Bad signal	13:56:00	53,98	-101 dBm	UMTS	UMTS	4054570	4057270	59 %
Bad handover	13:56:33	0	-105 dBm	UMTS	UMTS	4054570	4032050	74 %
Bad signal	14:59:47	55,8	-109 dBm	HSPA	HSPA	4004869	4004869	62 %
Bad handover	15:08:28	36,9	-85 dBm	UMTS	UMTS	3998982	4029490	50 %
Other reason	12:14:25	32,23	-69 dBm	UMTS	UMTS	4027990	4027990	57 %
Other reason	14:46:38	120,15	-85 dBm	HSPA	HSPA	4005742	4005742	37 %
Bad signal	10:27:47	64,83	-113 dBm	UMTS	UMTS	4037201	4037201	69 %
...	...	...	...	...	...	...	...	...

TABLE I
CALL DROP REASONS.

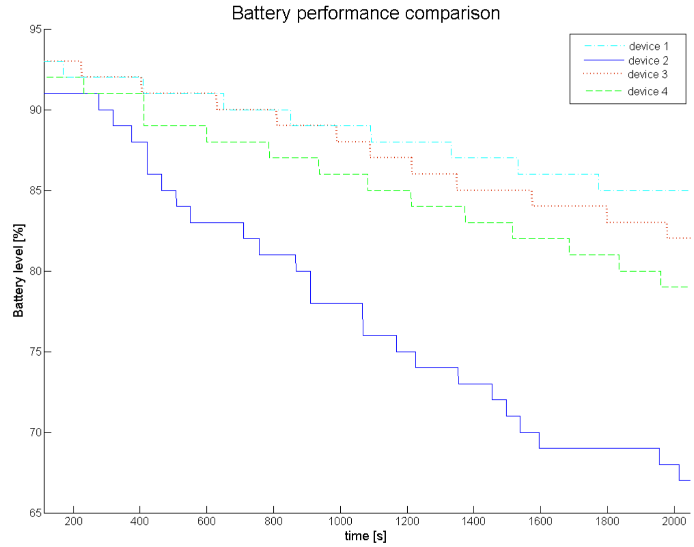

Fig. 5. Battery life comparison in 3G networks.

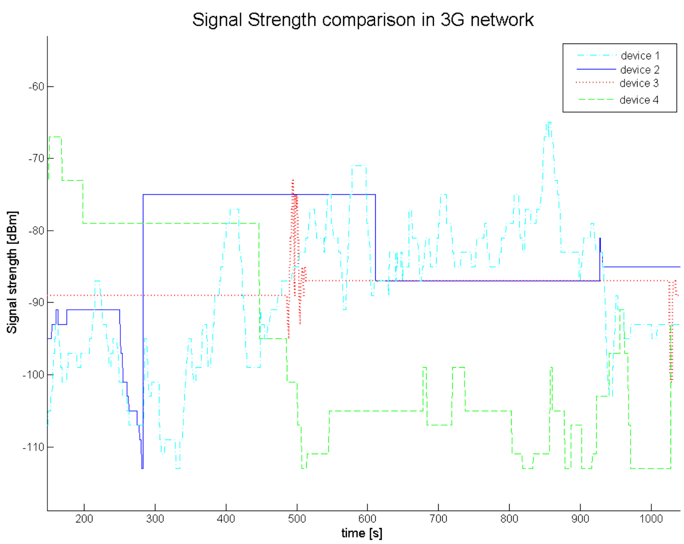

Fig. 4. Signal strength comparison in 3G networks

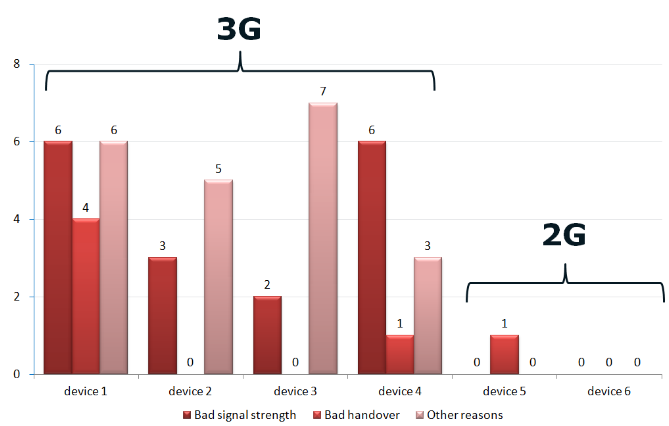

Fig. 6. Histogram of the detected call drops per device.

and further investigation will be needed. Indeed, these APIs do not provide any methods to read low level parameters directly from the internal modem. If the vendors had implemented the relative APIs, it would be possible to obtain many more parameters, using Android methods, compared to the ones we have been working with. These parameters could be the E_c/I_0 (RSCP divided by the total received power in the channel bandwidth), out of sync (UE lost frequency synchronization with the base station), TX (Transmitted) power and all the data that the internal modem of every phone is tracking. With this low level parameters we would be able to understand better where does the call drop issue come from: network or device.

Apart from confidential issues, we are not showing the brand of each phone as we did not perform enough days of drive test to extract some statistical conclusions about the performance of each phone. Also, as we can see in Figure 6, device5 just had one call drop and device6 had none, as expected, as the phones were working almost all the time in GSM. Indeed, GSM is an old and mature technology, which has been improved and refined over the last 20 years. $2G$ network coverage is also larger than $3G$ network, which still have some black zones especially in rural areas. As we can see in the bottom of Figure 3 several call drops were found outside Aalborg city area, because there was poor $3G$ coverage.

The network is not the only component in the advent of the call drop issue. This is also related with the design of the smartphones, particularly regarding the internal antenna, which usually has less sensitivity compared to old devices with an external one.

Despite of the fact that we managed to detect call drops, the aim was to develop an app that could work in both ends of a voice call, not just the scenario that we tested. However, there was no way to detect, from the programming perspective, when the user ended the call by purpose. The problem that we encountered in our research is that smartphones do not have a physical “red button”; the user is ending the call by touching the screen and is not possible to recognize this event with the app. This is due to security restrictions related with the dialling interface on Android. This problem can be furthermore studied, because finding a way of determining if the user pushed the *“red button”*, in a mobile-to-mobile call, will improve considerably the system.

Moreover, there is a specific class in Android for UMTS (Universal Mobile Telecommunications System) networks called *NeighboringCellInfo*, from which we can obtain the received signal strength and the CID parameters of the different base stations that are surrounding an UE. This class is implemented, but unfortunately, the *get* methods are working only in GSM. For UMTS and WCDMA (Wideband Code Division Multiple Access), the information that we want to acquire is treated as *not available*.

Using the other statistics, like in Figure 4, we can see how the signal strength is updated with a different frequency depending on each device. This is related with the sensitivity of the receiver of each phone or with the time interval at which the internal modem of the mobile phone sends the received data to the related API. This information is important, as we see the signal strength of device1 is varying the whole time, while device2 and device3 have intervals of several minutes where the signal strength is not changing. So we can conclude that we cannot trust 100% some information, like the signal strength, that the Android API is providing, as the real values may not be showed on the display.

V. CONCLUSION

The paper has described a high level approach for field measurements and call drops detection. We have demonstrated that a classification of call drops can be made with this approach.

After a post analysis of the measurement campaigns, and with a good understanding of the parameters variations, we managed to locate the call drops and express the specific moment when the call dropped. Furthermore, the reason behind a call drop is stated and the visualization on Google Maps of each type of call drop, along with important parameters, is performed. We proved that our detection system works for the considered case.

VI. FUTURE WORK

The app is a first version of a call drop analyzer based and different features can be improved in future work:

- implementation of the low level approach
- extension to a mobile-to-mobile system
- use of the Android class *NeighboringCellInfo*

Regarding the implementation of the low level approach, the vendors should implement the relative APIs, so that the user can get the parameters using Android methods. Following this approach, we would be able to obtain more results involving many more parameters that we cannot get at the moment.

As we mentioned in section IV, these parameters are measured by the relative modem present in the device, but they are not accessible directly by the users. However, these parameters can be accessed using a special trace tool, by sending AT commands to the internal modem, but for the moment, this can be done only through USB communication interface.

This approach is useful for doing a better classification of call drops and for developing a complex call drop analyzer.

The app is able to detect call drops only from a mobile-to-fixed line call, so finding a way of determining if the user ended the call by purpose, in a mobile-to-mobile call, would improve considerably the system.

The use of the specific Android class for UMTS networks called *NeighboringCellInfo* is not supported from different vendors. However, this class can be used in our app in the future if the vendors will make the APIs compatible with the methods. For this reason, new tests using future devices can be really useful to prove the features of the app.

REFERENCES

[1] P. Reichl, “From quality-of-service and quality-of-design to quality-of-experience: A holistic view on future interactive telecommunication services,” in *Software, Telecommunications and Computer Networks, 2007. SoftCOM 2007. 15th International Conference*, vol. 1, 2009, pp. 1–6.

[2] R. Kreher, *UMTS Performance Measurement. A practical guide to KPIs for the UTRAN environment*. Wiley, 2006.

[3] 3GPP, *Telecommunication management; Key Performance Indicators (KPI) for UMTS and GSM*, ch. Technical Specification 32.410.

[4] waze Mobile, “Waze homepage,” 2012. [Online]. Available: http://www.waze.com/

[5] I. Staircase 3, “Open signal map homepage,” 2012. [Online]. Available: http://opensignalmaps.com/

[6] Y. Yan, J. Yu, J. Wu, and S. Ma, “Design and implementation of a mobile gis for field data collection,” in *Computer Science and Information Engineering, 2009 WRI World Congress*, vol. 1, 2009, pp. 235–241.

[7] T. Kamei and T. Shima, “Study on advanced smartphone use of probe traffic information,” in *Communication Technology and Application (ICCTA 2011), IET International Conference*.

[8] J. Muñoz, B. Coll-Perales, and J. Gozalvez, “Research testbed for field testing of multi-hop cellular networks using mobile rays,” in *Local Computer Networks (LCN), 2010 IEEE 35th Conference*.

[9] H. Li and L. Zhijian, “The study and implementation of mobile gps navigation system based on google maps,” in *Computer and Information Application (ICCIA), 2010 International Conference*.

[10] R. K. Ganti, S. Srinivasan, and A. Gacic, “Multisensor fusion in smartphones for lifestyle monitoring,” in *Body Sensor Networks (BSN), 2010 International Conference on*, 2010, pp. 36–43.

[11] G. Boggia, P. Camarda, A. D’Alconzo, A. De Biasi, and M. Siviero, “Drop call probability in established cellular networks: from data analysis to modelling,” in *Future Networks, 2010. ICFN ’10. Second International Conference*, vol. 5, 2005, pp. 2775–2779.

[12] Y. Iraqi and R. Boutaba, “Handoff and call dropping probabilities in wireless cellular netowrks,” in *Wireless Networks, Communications and Mobile Computing, International Conference*, vol. 1, 2005, pp. 209–213.

[13] R. Zhao, X. Wen, D. Su, and W. Zheng, “Call dropping probability of next generation wireless cellular netowrks with mobile relay station,” in *Future Networks, 2010. ICFN ’10. Second International Conference*.

[14] K. Sudhindra and V. Sridhar, “Root cause detection of call drops in live gsm network,” in *TENCON 2011 - 2011 IEEE Region 10 Conference*.

[15] S. Seytnazarov, “Rf optimization issues in gsm networks,” in *Application of Information and Communication Technologies (AICT), 2010 4th International Conference on*, 2010, pp. 1–5.

[16] N. D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. T. Campbell, “A survey of mobile phone sensing,” *IEEE Communications Magazine*, pp. 140 – 150.

[17] S. S. Kanhere, “Participatory sensing: Crowdsourcing data from mobile smartphones in urban spaces,” in *2011 12th IEEE International Conference on Mobile Data Management*, vol. 2, pp. 3 – 6.